FPT

2 novembre 2021 13:24

Pb NP-complet → **Approx** : $O(n^c)$, $APP \leq c \cdot OPT$

Pas d'algo $O(n^c)$ → **Complexité paramétrée** : FPT, Solution exacte, $O(2^k \cdot n^c)$, $O(2^k \cdot poly(n))$ où k est un paramètre petit

Revenons à Vertex-Cover (Entrée : $G=(V,E)$; Sortie : $X \subseteq V$ qui couvre E)

Si on veut une sol optimale

```
algoNaïf( G=(V,E) )
    X = V
    pour S ⊆ V              // n = |V|
        si S couvre E et |S| < |X|       // m = |E|
            X = S
    return X
```

2^n itérations (pour) ; $O(m)$ (si)

$O(2^n \cdot m)$ Trop lent

Supposons qu'on a un param. $k \in \mathbb{N}$ tel que
- si $OPT > k$, pas intéressé
- si $OPT \leq k$, je veux le savoir

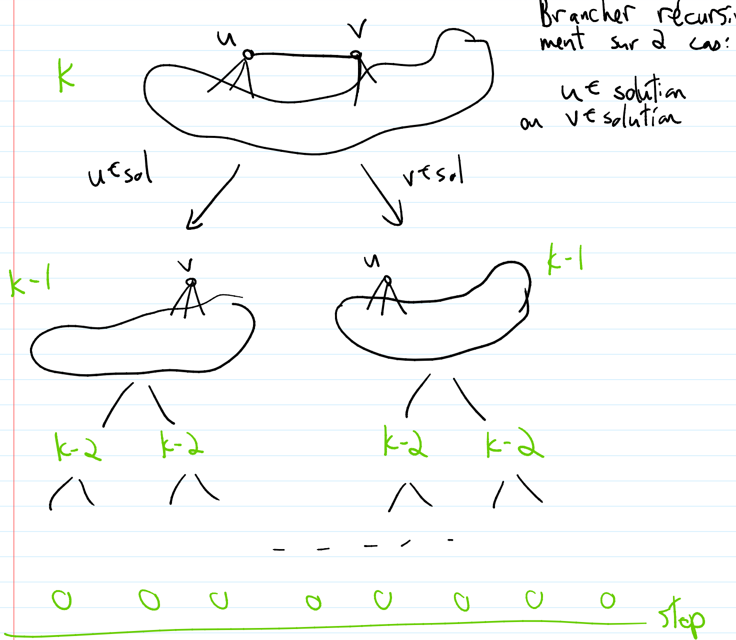
k-Vertex-Cover $\quad O(2^k \cdot (n+m))$

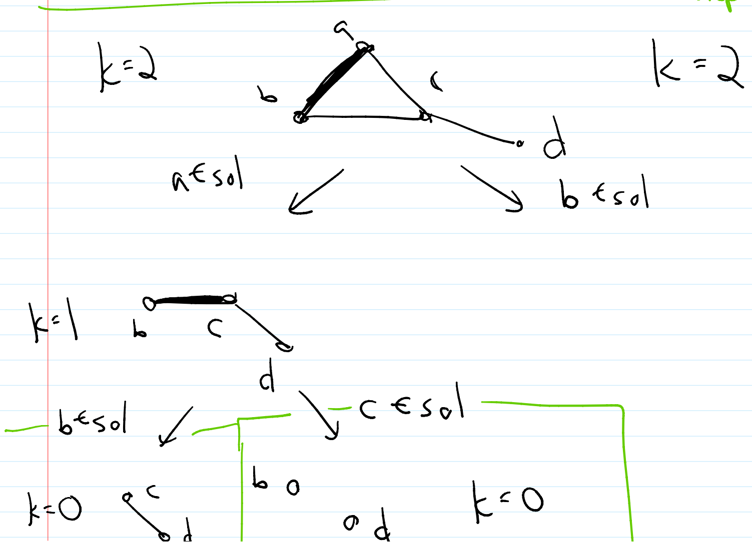
Entrée : $G=(V,E)$

Paramètre : k

Sortie : true s'il $\exists$ un vertex cover X t.q. $|X| \leq k$, false sinon

Brancher récursivement sur 2 cas : $u \in$ solution ou $v \in$ solution



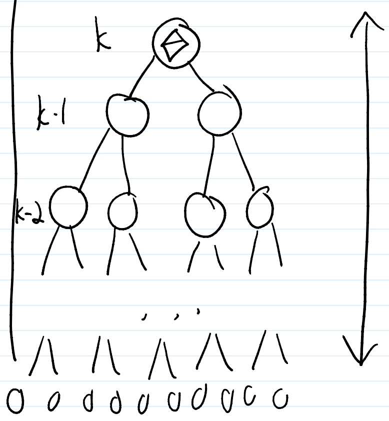


$k=0$ stop | oui, return true $k=0$

```
vcFPT(G, k)
  si |E| = 0, return true
  si k = 0, return false
  Soit uv ∈ E (arbitraire)
  G_u = copie de G après avoir enlevé u
  si vcFPT(G_u, k-1)
     return true
  G_v = copie de G sans v
  si vcFPT(G_v, k-1)
     return true
  return false
```

Complexité

Cet algo fait un arbre de récursion où
- chaque noeud = un appel récursif
- racine = appel initial



niv 0 : $2^0 = 1$

niv 1 : $2^1 = 2$

niv 2 : $2^2 = 4$

niv k : $2^k = 2^k$

$1+2+4+8+\ldots+2^k$

On a un arbre binaire de hauteur $\leq k$

$\Rightarrow$ #noeuds = #d'appels $\leq 2^{k+1}-1$

Chaque noeud prend un temps $O(n+m)$

$\Rightarrow$ Complexité : $O((2^{k+1}-1)(n+m))$

$O(2^k(n+m))$

$1+2+4+8+\ldots+2^k$

$= 2^k + 2^{k-1} + \ldots + 4 + 2 + 1$

binaire 1 1 1 ... 1 1 1) +1

2^{k+1} en bin. 1 0 0 0 ... 0 0 0

$\Rightarrow \sum_{i=1}^{k} 2^i = 2^{k+1} - 1$

Si on veut le k minimum

```
pour k = 1..n
   si vcFPT(G, k)
      return k
```

Définition

Soit P un pb algorithmique. Soit I une instance de P et k un paramètre. On dit que (I,k) est une instance paramétrée de P.

On dit que P est résoluble à paramètre fixe en fct de k (fixed-parameter tractable, ou FPT) s'il $\exists$ un algo A t.q. $\forall$ instance param. (I,k) de P,
- retourne toujours une solution correcte pour (I,k)
- A s'exécute en temps $O(f(k)\, n^c)$ où // $n = |I|$

  $f(k)$ est n'importe quelle fct de k seulement et c = constante indépendante de k

FPT	Pas FPT
$O(2^k n)$	$O(n^k)$
$O(k! \cdot n^2)$	$O(k \cdot 2^n)$

$O(k! \cdot n^2)$

$O(\underbrace{k^{100} \cdot \log_k 2^k \cdot 1000^k}_{f(k)} \cdot 2n)$

$O(2^{2^{2^{2^{2^{2^k}}}}} \cdot n)$

$O(k \cdot 2^n)$

$O(k\, n^{1/k})$

Exercice: MAX-CLIQUE
algo $O(2^k \cdot n^c)$ ou k = degré max du graphe
$k = \max_{v \in V} |N(v)|$

Pour en finir avec vertex-cover

```
pasFPTvc(G, k)
    pour i = 1..k
        pour S ⊆ V(G) tel que |S| = i
            si S couvre |E|
                return S
    return false
```

Complexité:

de sous-ens. énumérés

$$\sum_{i=1}^{k} \binom{n}{i} \geq \binom{n}{k}$$

$$= \frac{n!}{k!(n-k)!}$$

$$= \frac{n(n-1)(n-2)\cdots(n-k+1)}{k!}$$

On suppose que $n \gg k$

$\geq ?$

$$\geq \frac{(n-k)^k}{k!} = \frac{n^k - kn^{k-1} + \dots + k^k}{k!}$$

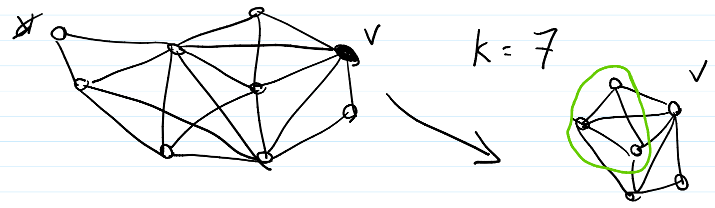
$$\in \Omega\left(\frac{n^k}{k!}\right)$$

Pas FPT car pas la forme
$O(f(k)n^c)$

MAX-CLIQUE

Entrée: graphe $G = (V, E)$
Param.: $\deg(G) = \max |N(v)| = k$

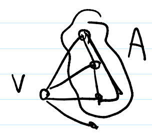
Entrée: graphe $G=(V,E)$
Param.: $\deg(G) = \max_{v \in V} |N(v)| = k$
Sortie: une clique de taille maximum de G



Idée: pour chaque v
trouver la + grosse clique qui contient v

maxCliqueDeg($G=(V,E)$, k)

si $\max_{v \in V} |N(v)| > k$, return "Error"



$X = \emptyset$

n iter. — pour $v \in V$

$\leq 2^k$ iter. — pour chaque $A \subseteq N(v)$

$\leq k^2$ — si A est une clique et $|A \cup \{v\}| > |X|$ → $O(|A|^2)$ pour vérifier qu'il y a une arête entre chaque paire

$X = A \cup \{v\}$

return X

Complexité: $O(n \cdot 2^k \cdot k^2) = O(\underbrace{k^2 \cdot 2^k}_{f(k)} \cdot n)$

$f(k) n^1$

Autre paramétrisation

Entrée: graphe G
Param: k = taille de la clique voulue
Sortie: true s'il $\exists$ une clique de taille k ou plus,
false sinon

Pas d'algo FPT connu.
Ce pb est W[1]-complet.
↳ analogue de NP-complet en FPT
↳ il n'existe probablement pas d'algo $O(f(k) n^c)$